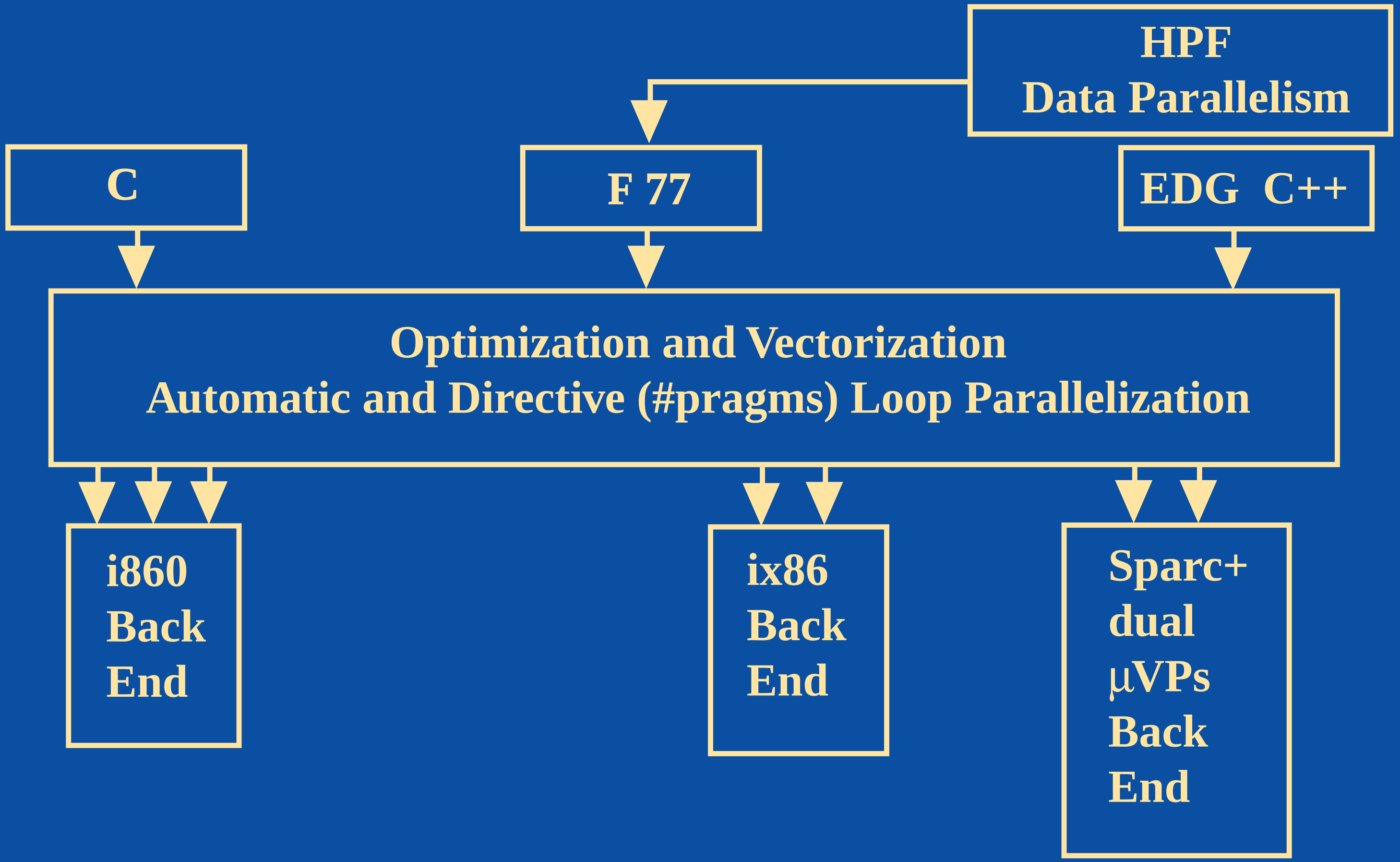
# Parallel Software Development Tools For Pentium Pro Systems

The Portland Group, Inc. (PGI)
http://www.pgroup.com

Vincent Schuster - CEO
vinces@pgroup.com

# pre-1997 Unix Compilers



HPF
Data Parallelism

C

F 77

EDG C++

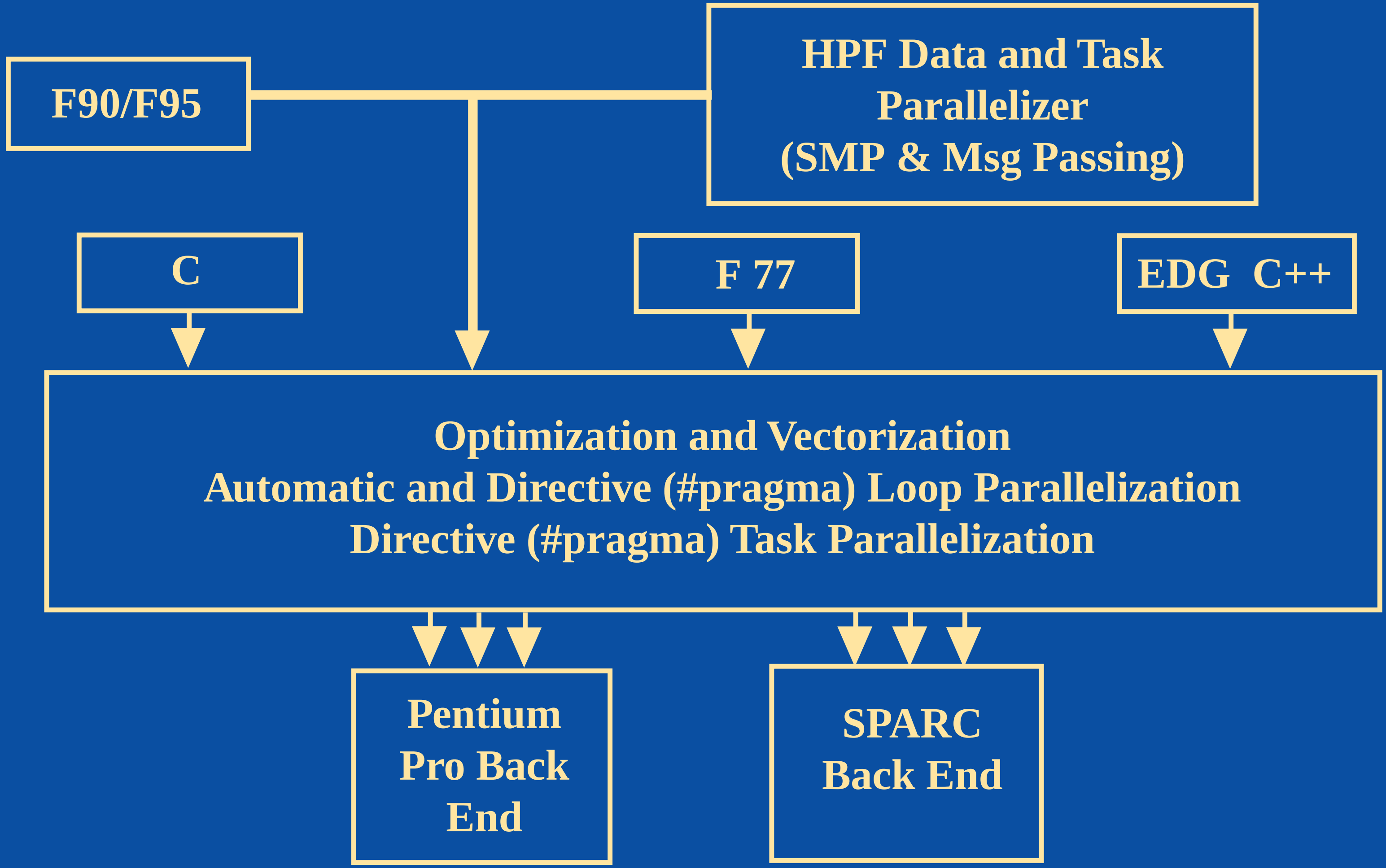
Optimization and Vectorization
Automatic and Directive (#pragms) Loop Parallelization

i860
Back
End

ix86
Back
End

Sparc+
dual
µVPs
Back
End

# 1997/1998 Unix & NT Compilers


F90/F95
HPF Data and Task Parallelizer (SMP & Msg Passing)
C
F 77
EDG C++
Optimization and Vectorization
Automatic and Directive (#pragma) Loop Parallelization
Directive (#pragma) Task Parallelization
Pentium Pro Back End
SPARC Back End

The Portland Group, Inc. (PGI)

# HPF 2.0 Core

- Data parallelism can readily be expressed.
- Well-expressed data mapping hints can help significantly.
- F95 extensions: FORALL, PURE

# HPF 2.0 Extensions

- non-Task Parallel Extensions
  - **ON HOME, RANGE, SHADOW, GEN_BLOCK, INDIRECT, EXTRINSICS**
  - **Asynchronous I/O**
- Task Parallel Extensions
  - **ON, RESIDENT, TASK_REGION, notion of ACTIVE PROCESSOR sets**

# PGHPF Compiler RoadMap

- ♠ PGHPF 2.2 (March 1997)
  - ♦ **InterProcedural Analysis, HPF/CRAFT, SMP HPF.**
- ♠ PGHPF 3.0 (Later 1997)
  - ♦ **Native Pentium Pro HPF Compilers, Parallel I/O, Auxiliary HPF Libraries, non-Task Parallel HPF 2.0 Extensions, improve handling of irregular and sparse data (Unix and NT)**
- ♠ PGHPF 3.x (Mid 1998)
  - ♦ **Task Parallel HPF 2.0 Extensions, Asynchronous I/O**

# SMP HPF

- Each CPU is treated as a separate PROCESSOR node
  - **Native and Non-native implementations**
- Direct memory accesses where beneficial (e.g., indirect arrays)
  - **Data aggregation and vectorization and indirect array schedule creations and reuse often are still useful. Decisions can be made within compiler based on system parameters.**
- Sequence and storage association relaxed
  - **Arrays are partioned in memory - substantial reductions in false sharing.**
- Expect to meet or exceed MPI performance in 1997.

# DSM HPF

- Each CPU is treated as a separate PROCESSOR or node.
- Data movement between nodes is via low level one-way communications, e.g., put( ) and get ( ).
- Cray T3-E Example
    - **Exceeds stringent performance requirements for specified benchmarks**
        - **Hydro-dynamics codes**
        - **Key nuclear simulation codes and kernels**
        - **PUT ( ), GET ( ) one-way communication mechanisms**
        - **Lots of irregular array accesses**
        - **Scaling to 256 nodes**
- SGI Origin and HP Exemplar are prime candidates.

# Hybrid Memory

- Each node consists of multiple SMP CPUs.
- Access between nodes is accomplished better than other data movement mechanisms.
- Data locality within each node is ensured due to HPF directive hints.
- HPF treats each node as a separate “PROCESSOR.”
- SMP parallelization is used within each node using SMP compilation techniques.

# Teragon Hybrid

- PGHPF compiles to move data between nodes. PGF77 SMP parallelized the resultant output code.
- So far, PGI has run one example - SHALLOW. It has scaled linearly up to 1085 nodes (2170 CPUs), uses threads to parallelize loops between each pair of node processors.

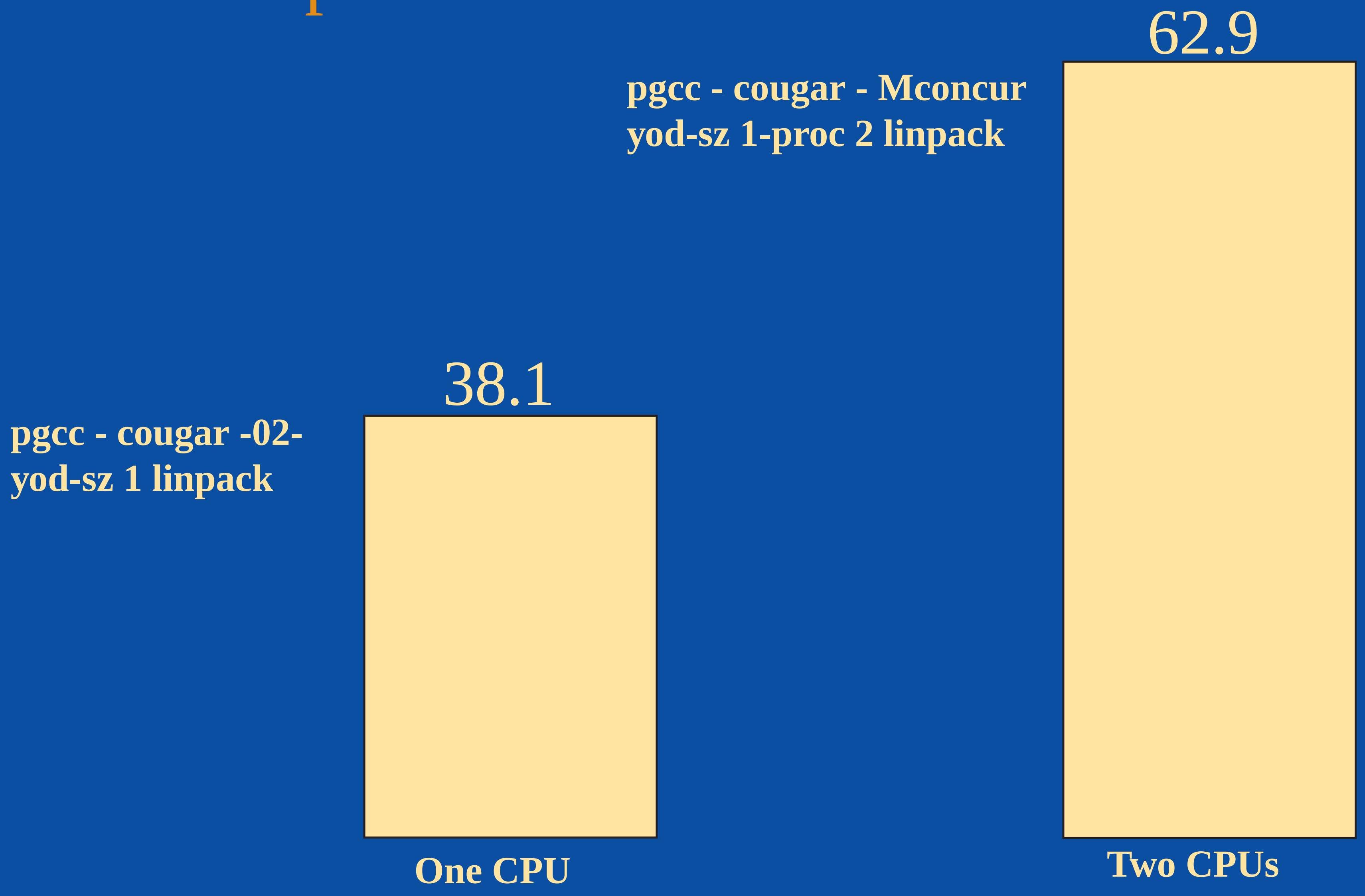
# SMP Node Compiler Features

- C, C++ and F77
  - **Written in portable C; ported to a multitude of platforms (Unix and DOS)**
- In 1997, F90 and F95 will be native for Pentium Pro and Sparc
- Optimizer, Vectorizer, SMP Parallelizer, and Code Gen are shared by all the languages.
- Linux (threadless), Windows NT (later 97), Solaris86,…
- PCF-style directives & #pragmas across languages
  - **Other spellings in the pipeline**
  - **Unified Parallel Programming Model is a goal**

# HPF and SMP Parallel Tool Support

- Existing
  - PGPROF - Post-mortem profiler (message passing statistics)
  - syPABLO - Dan Reed @ U of Illinois
  - AIMS - NASA Ames
  - HPF/MPI Binding - Ian Foster @ Argonne facilitates task parallelization
  - TotalView Debugger - Dolphin (ex BBN debugger)
- 1997+
  - Pentium Pro Parallel Debugger Support
  - PGPROF - Scaling statistics and prediction
  - UNIX and NT Tool support

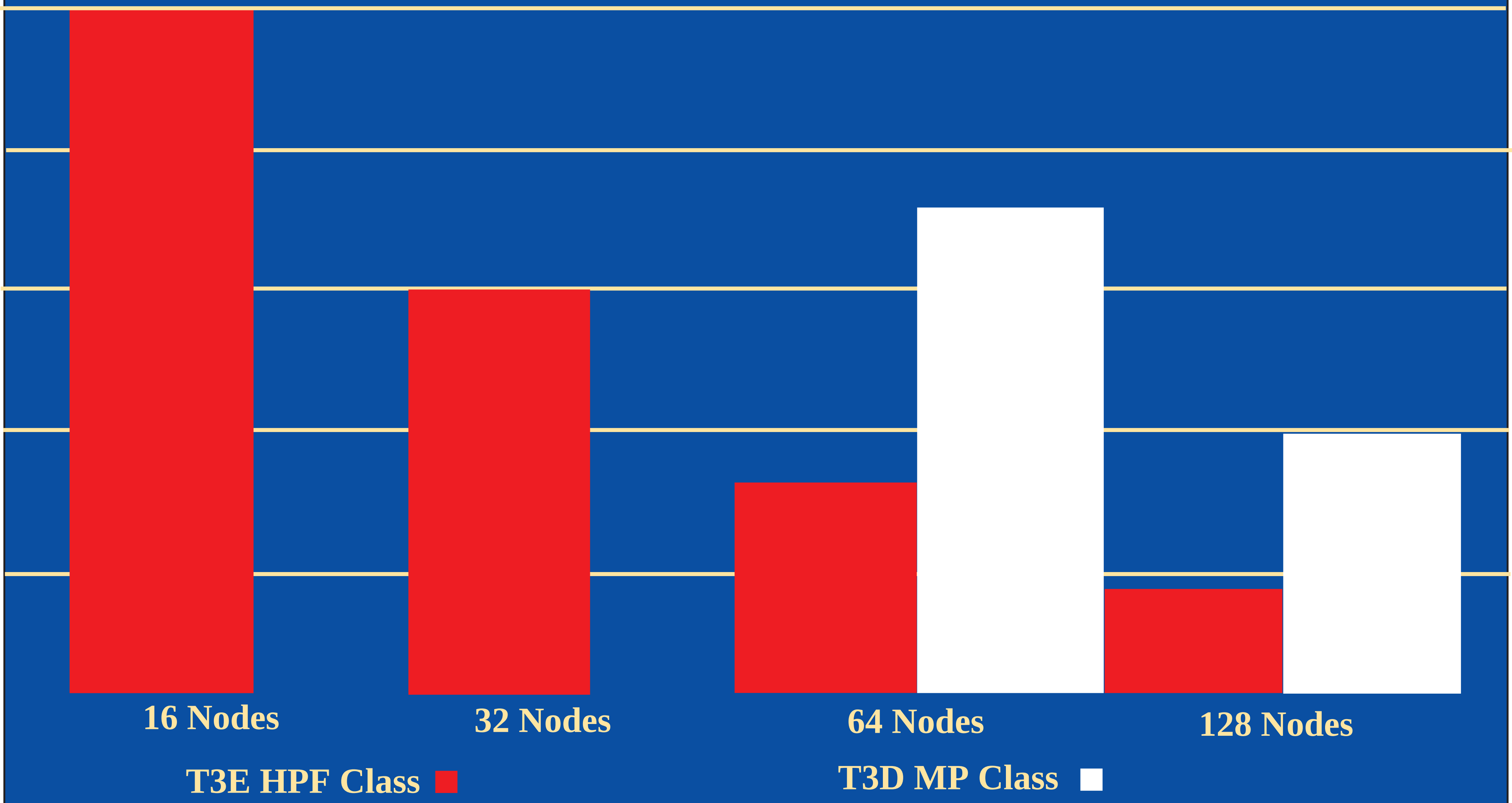
# Linpack Double Precision



pgcc - cougar -02-
yod-sz 1 linpack

pgcc - cougar - Mconcur
yod-sz 1-proc 2 linpack

# T3-E BT NPB Class B: HPF and MPI



# Migration Path Details

- ♠ UNIX developers migrating to NT
  - ♦ **Many try Linux or Solaris86 first**
  - ♦ **Want 2-way bridge between parallel RISC and PPro Unix platforms**

    **Similar software development toolset would enhance code portability and preserve their existing expertise equity.**
  - ♦ **Predictive performance monitoring tools would be welcome.**

    **Predict performance and portability.**
  - ♦ **This appears to be the earlier migration path.**

- ♠ NT developer migrating to Parallel NT
  - ♦ **IDE, simple parallel prog model & tools, use NT & classes.**

- ♠ UNIX to NT and Parallel UNIX to Parallel NT
  - ♦ **May be most flexible to change but would like to have same SW design toolset across both platforms if they intend on continuing to support codes across both platforms.**

# Java

- Is there an effective web strategy targeting an efficient parallel programming model targeted at HPC users and leveraging the JVM and Java technologies?